



PENERAPAN ALGORITMA A* DALAM PENYELESAIAN RUTE TERPENDEK PENDISTRIBUSIAN BARANG

Rosita Ayu Nugraeni✉, Mulyono, Rochmad.

Jurusan Matematika, FMIPA, Universitas Negeri Semarang, Indonesia
Gedung D7 lantai 1 Kampus Sekaran, Gunungpati, Semarang, 50229

Info Artikel

Sejarah Artikel:
Diterima Januari 2014
Disetujui Februari 2014
Dipublikasikan Mei 2015

Keywords :
Algoritma A*;
Pendistribusian Barang;
Rute Terpendek

Abstrak

Penelitian ini mengkaji sebuah permasalahan pencarian solusi untuk masalah penentuan rute terpendek pendistribusian barang. Tujuan penelitian ini adalah untuk mengetahui dasar-dasar algoritma A*, untuk meneliti penentuan rute terpendek pendistribusian barang dengan algoritma A*, dan untuk meneliti penentuan rute terpendek pendistribusian barang diaplikasikan dengan berbantuan *software*. Pengambilan data dilakukan dengan cara mendokumentasikan data di kantor CV Mitra Adi Busana Semarang, selanjutnya dilakukan pencarian jarak dengan bantuan *Google Maps*. Analisis data dilakukan dengan menggunakan algoritma A* yang kemudian diaplikasikan dengan *software Visual Basic.Net*. Dari analisis yang dilakukan dengan cara manual maupun berbantuan *software*, diperoleh rute terpendek pendistribusian barang yaitu, CV Mitra Adi Busana - Hotel Gumaya - Hotel Merbabu - Hotel Novotel - Hotel Ibis - Hotel Grand Saraswati - Hotel Royal Phoenix - Hotel Neo Candi - Hotel Plaza dengan panjang rute terpendek 19595 meter. Dari hasil analisis dengan algoritma A* dan berbantuan *software* diperoleh rute terpendek yang sama dalam pendistribusian barang untuk mencapai seluruh lokasi pendistribusian barang dengan rute terpendek yang minimal.

Abstract

This study examine a problem of finding solutions to the problem of determining the shortest route distribution of goods. The purpose of this study is to determine the basics of the A* algorithm, to examine the determination of the shortest route distribution of goods with A* algorithm, and to examine the determination of the shortest route distribution of goods applied to the assisted software. Data collection is done by documenting the data in the office of the CV Mitra Adi Busana Semarang, then performed a search distance with the help of Google Maps. Data analysis is done by using the A* algorithm then applied with Visual Basic.Net software. From the analysis done by means of manual or assisted softwares, distribution of goods obtained by the shortest route, namely, CV Mitra Adi Busana - Gumaya Hotel - Merbabu Hotel - Novotel Hotel - Ibis Hotels - Grand Saraswati Hotel - Royal Phoenix Hotel - Neo Candi Hotel - Plaza Hotel with a distance shortest route of 19595 metres. From the analysis with A* algorithm and assisted softwares in order to obtain the shortest route in the distribution of goods to reach all locations of distribution of goods with minimal shortest route.

Pendahuluan

Transportasi telah menjadi salah satu kebutuhan penting dalam kegiatan sehari-hari di kehidupan bermasyarakat. Kemajuan teknologi informasi yang ada sekarang, dapat digunakan sebagai sarana untuk meningkatkan pelayanan umum, di antaranya para pengguna sarana dapat memperoleh informasi lalu-lintas dengan cara yang mudah.

Menurut Setyawan (2012), permasalahan distribusi adalah bagian dari permasalahan logistik yang didefinisikan sebagai penyediaan barang dan jasa dari titik persediaan ke titik permintaan. Persoalan distribusi terdapat di berbagai bidang pelayanan umum, misalnya persoalan pengangkutan sampah, persoalan jemputan penumpang bus, persoalan pengiriman barang, dan penentuan jalur pembersihan jalan. Dalam permasalahan distribusi, penentuan rute merupakan permasalahan yang cukup vital. Dengan penentuan rute yang baik sebuah armada dapat melaksanakan tugas distribusi dengan optimal.

CV Mitra Adi Busana merupakan salah satu perusahaan yang bergerak dalam bidang binatu. CV Mitra Adi Busana sendiri memiliki konsumen yang ada di kota Semarang. Dalam mengirimkan barang dari pusat ke pelanggan di berbagai tempat, perlu adanya suatu sistem yang mampu meminimalisasi rute pengiriman barang. Permasalahan itu merupakan masalah model jaringan yang sama dengan permasalahan rute terpendek.

Pencarian rute terpendek adalah usaha untuk mencari rute yang paling dekat dari posisi awal hingga akhir dengan beban paling ringan atau sedikit dibandingkan dengan seluruh rute yang ada. Terdapat banyak algoritma yang dapat digunakan dalam pencarian rute terpendek. Salah satunya adalah algoritma A*.

Menurut Ananda (2010), algoritma A* pertama kali ditemukan pada tahun 1968 oleh Peter Hart, Nils Nilsson dan Bertram Raphael. Dalam tulisan mereka, algoritma ini dinamakan algoritma A. Penggunaan algoritma ini dengan fungsi heuristik yang tepat dapat memberikan hasil yang optimal, maka algoritma ini disebut A*.

Algoritma A* (*A Star*) adalah algoritma pencarian rute terpendek yang merupakan perbaikan dari algoritma *Best First Search* (BFS) dengan memodifikasi fungsi heuristiknya (Kusumadewi, 2003: 43). Seperti halnya pada BFS, untuk menemukan solusi, A* juga

dituntun oleh fungsi heuristik, yang menentukan urutan titik mana yang akan dikunjungi terlebih dahulu. Heuristik merupakan penilai yang memberi harga pada tiap titik yang memandu A* mendapatkan solusi yang diinginkan.

Menurut Gusyanto (2012), algoritma A* mengevaluasi titik dan menggabungkan $g(n)$ yaitu jarak untuk mencapai titik dan $h(n)$ yaitu jarak yang diperlukan dari titik untuk mencapai tujuan, sehingga:

$$f(n)=g(n)+h(n)$$

Jarak dari titik awal ke *node n* adalah $g(n)$ dan $h(n)$ adalah perkiraan jarak terpendek dari titik *n* ke titik tujuan. Sedangkan $f(n)$ adalah perkiraan solusi dengan jarak terpendek melalui *n*.

Dengan demikian, untuk menemukan solusi terbaik, hal pertama yang dicoba adalah titik dengan nilai $g(n)+h(n)$ terendah. Strategi ini jelas lebih baik dengan disediakannya nilai heuristik $h(n)$ yang dapat memenuhi kondisi tertentu sehingga A* menjadi optimal.

Menurut Supriyani (2008), berikut adalah langkah-langkah pencarian rute terpendek dengan algoritma A*:

1. Dimulai dengan *Open List* yang berisi titik awal. Nilai g dari *node* tersebut menjadi 0 dan nilai h sesuai nilai heuristik. *Close list* berupa list kosong.
2. Hingga *node* tujuan ditemukan, ulangi langkah berikut ini:

Jika tidak ada titik pada *Open List* maka laporkan kekeliruan. Jika tidak, ambil *node* pada *Open List* yang memiliki nilai f terendah dan namakan sebagai *best node*. Pindahkan dari *Open List*, masukkan ke *Close List*. Untuk setiap *successor*, lakukan langkah-langkah berikut ini:

 - a) Set *successor* menunjuk kembali ke *best node*. Link ke belakang ini memungkinkan jalur yang terbentuk dipulihkan setelah hasilnya ditemukan.
 - b) Hitung $g(\text{successor})=g \text{ best node}+\text{biaya}$ yang diperlukan untuk mencapai hasil dari *best node* ke *successor*.
 - c) Jika *successor* berada di *Open List* maka namakan sebagai *old. Parent* dari *old* harus diubah apabila lintasan yang baru saja

ditemukan menuju *successor* lebih baik dari lintasan yang ada sekarang.

d) Jika *successor* ada di *Close List*, maka namakan sebagai *Close List old* dan tambahkan ke *successor* dan *best node*. Periksa apakah lintasan yang baru ini lebih baik dari langkah sebelumnya, dan set link *parent* bila *g* dan *f*. Jika lintasannya lebih baik, maka dilakukan perbaikan pada *successor old* dengan pergerakan secara *dept first* dimulai dari *old*, ubah nilai *g* dan *f* tiap *node*, akhiri tiap cabang bila telah mencapai lintasan yang lebih baik dari sebelumnya. Tiap link *parent* dari *node* menunjuk ke *parent* terbaik. Jika *node*-nya menuju ke titik awal lanjutkan penyebaran. Jika tidak maka nilai *g*-nya mencerminkan lintasan yang lebih baik dan penyebaran dihentikan. Tetapi mungkin dengan nilai baru *g* yang *node*-nya disebarakan ke bawah menjadi lintasan yang lebih baik dari lintasan sebelumnya.

e) Jika *successor* tidak pernah berada di *Open List* atau *Close List*, maka masukkan ke *Open List* dan tambahkan ke daftar *successor* dari *best node*.

Menurut Hakim (2012), Visual Basic.Net (VB.Net) merupakan pengembangan dari Microsoft Visual Basic versi sebelumnya. Kata “Visual” menunjukkan cara yang digunakan untuk membuat *Graphical User Interface* (GUI). Dengan cara ini, tidak perlu lagi menuliskan instruksi pemrograman dalam kode-kode baris hanya untuk membuat sebuah desain form atau aplikasi. Tetapi dengan sangat mudah yakni cukup melakukan drag dan

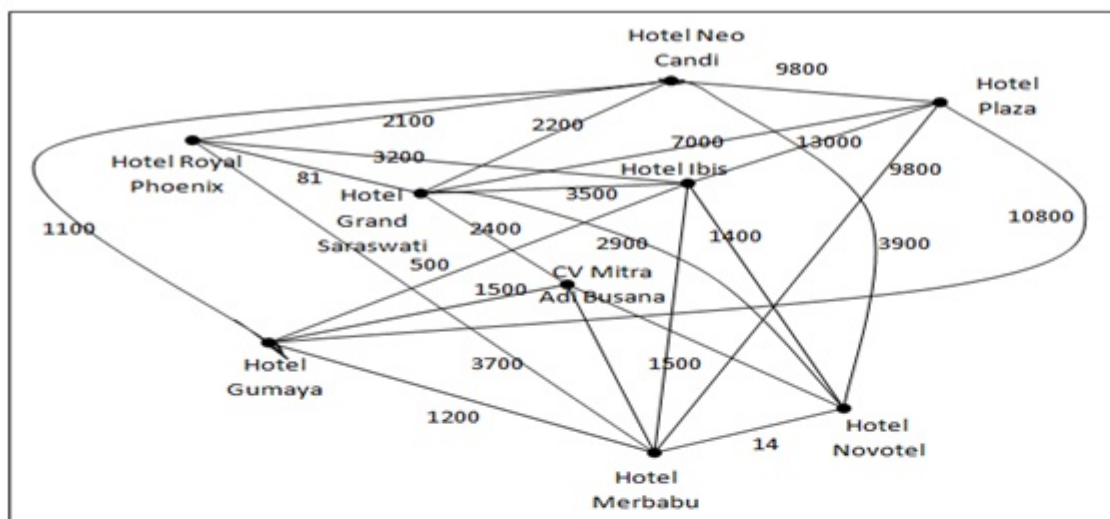
drop objek-objek yang akan digunakan. VB.Net dapat dijadikan alat bantu untuk membuat berbagai macam program komputer. Aplikasi VB.Net hanya dapat dijalankan pada sistem Operasi Windows. Aplikasi yang dapat dihasilkan dengan bahasa pemrograman. VB.Net adalah sistem aplikasi bisnis, *software* aplikasi SMS, *software* aplikasi *chatting*, permainan (*game*), dan lain-lain.

Metode Penelitian

Metode penelitian yang digunakan adalah metode studi kasus, di mana peneliti melakukan penelitian untuk mengumpulkan data-data yang diperlukan untuk penelitian ini. Sumber data diperoleh dari kantor CV Mitra Adi Busana Semarang dan diperoleh jarak antar lokasi dengan bantuan *Google Maps*, selain itu mengumpulkan beberapa literatur yaitu berupa jurnal, buku, dan literatur ilmiah lainnya yang mendukung penelitian ini. Metode yang digunakan adalah metode analisis hasil penyelesaian masalah penentuan rute terpendek pendistribusian barang dengan bantuan *software* Visual Basic.Net.

Hasil Penelitian dan Pembahasan

Proses pencarian rute terpendek menggunakan algoritma A* akan dijelaskan pada kasus di bawah ini dengan 9 tempat pendistribusian barang. Berikut adalah gambar pendistribusian barang CV Mitra Adi Busana dalam satuan meter. Graf yang menggambarkan jalur pendistribusian barang terdapat pada Gambar 1.



Gambar 1. Graf yang menggambarkan rute pendistribusian barang.

Jarak Euclidian merupakan jarak garis lurus dari masing-masing titik menuju titik tujuan, maka didapat $h(n)$ masing-masing titik. Pada permasalahan ini titik tujuan adalah titik I sebagaimana tersaji dalam Tabel 1.

Tabel 1 Jarak Euclidian tempat tujuan distribusi (dalam satuan meter)

A	B	C	D	E	F	G	H	I
6580	5210	5300	6800	7550	7600	7970	5000	0

Keterangan Tabel 1:

- A= CV Mitra Adi Busana
- B= Hotel Grand Saraswati
- C= Hotel Royal Phoenix
- D= Hotel Ibis
- E= Hotel Novotel
- F= Hotel Merbabu
- G= Hotel Gumaya
- H= Hotel Neo Candi
- I= Hotel Plaza

Berdasarkan graf yang telah ada, dapat diaplikasikan ke dalam perhitungan algoritma A*. Proses pencarian dengan algoritma A* adalah dengan menjumlahkan antara jarak sebenarnya dengan jarak perkiraan, dalam notasi matematika dituliskan $f(n) = g(n) + h(n)$. Proses pencarian dimulai dari titik A yaitu CV Mitra Adi Busana sebagai titik awal, sehingga diperoleh nilai $g(A)=0$, sementara nilai $g(A)$ pada setiap titik yang akan telah tersaji pada gambar di atas. Sebagai contoh berikut adalah pencarian jarak dengan algoritma A* dari titik awal A dan titik akhir I dengan melewati seluruh titik yang ada.

1. Fungsi Evaluasi (Melalui A):

$$f(B)=g(A)+g(A \text{ ke } B)+h(B)=0+2400+5210=7610^*$$

$$f(E)=g(A)+g(A \text{ ke } E)+h(E)=0+1615+7550=9265$$

$$f(F)=g(A)+g(A \text{ ke } F)+h(F)=0+1600+7600=9200$$

$$f(G)=g(A)+g(A \text{ ke } G)+h(G)=0+1500+7970=9470$$

Pada OPEN hanya terdapat satu titik, yaitu A. Titik A terpilih sebagai *best node* dan dipindahkan ke CLOSE. Bangkitkan semua *successor* A yaitu B, E, F, dan G. Keempat titik tersebut tidak ada di OPEN maupun CLOSE, maka keempatnya dimasukkan ke OPEN. Langkah ini menghasilkan OPEN=[B, E, F, G]

dan CLOSE=[A]. Titik B dengan bobot terkecil yaitu 7610 terpilih sebagai *best node* dan dipindahkan ke CLOSE.

2. Fungsi Evaluasi (Melalui A dan B):

$$f(E)=9265$$

$$f(F)=9200$$

$$f(G)=9470$$

$$f(C)=g(B)+g(B \text{ ke } C)+h(C)=2400+81+5300=7781^*$$

$$f(D)=g(B)+g(B \text{ ke } D)+h(D)=2400+3500+6800=11300$$

$$f(H)=g(B)+g(B \text{ ke } H)+h(H)=2400+2200+5000=9600$$

$$f(I)=g(B)+g(B \text{ ke } I)+h(I)=2400+7000+0=9400$$

Titik E melalui *parent* A dan B

$$f(E)=g(B)+g(B \text{ ke } E)+h(E)=2400+2900+7550=13850$$

Successor B dibangkitkan, yaitu C, D, E, H, dan I. Titik C, D, H, dan I belum pernah berada di OPEN maupun CLOSE, maka keempat titik tersebut dimasukkan ke OPEN. Sementara titik E sudah berada di OPEN, sehingga harus dilakukan pengecekan *parent* titik E perlu diganti atau tidak. Berdasarkan hasil perhitungan di atas *parent* titik E melalui A dan B lebih besar daripada melalui titik A. Dengan demikian, *parent* titik E tidak perlu diganti. Langkah ini menghasilkan OPEN=[E, F, G, C, D, H, I] dan CLOSE=[A, B]. Titik C dengan bobot terkecil dipilih sebagai *best node* dan dimasukkan ke CLOSE.

3. Fungsi Evaluasi (Melalui A, B, dan C):

$$f(E)=9265$$

$$f(F)=9200^*$$

$$f(G)=9470$$

$$f(D)=11300$$

$$f(H)=9600$$

$$f(I)=9400$$

Titik D, F, dan H melalui *parent* A, B, dan C

$$f(D)=g(C)+g(C \text{ ke } D)+h(D)=2481+3500+6800=12781$$

$$f(F)=g(C)+g(C \text{ ke } F)+h(F)=2481+3700+7550=13731$$

$$f(H)=g(C)+g(C \text{ ke } H)+h(H)=2481+2100+5000=9581$$

Successor C dibangkitkan, yaitu D, F, dan H. Titik D, F, dan H sudah pernah berada di OPEN, maka perlu dilakukan pengecekan *parent* dari masing-masing titik. Berdasarkan perhitungan di atas *parent* dari titik H perlu diganti karena bobotnya lebih kecil daripada bobot A ke B, sementara *parent* dari titik D dan F tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini

menghasilkan $OPEN=[E, F, G, D, H, I]$ dan $CLOSE=[A, B, C]$. Titik F dengan bobot terkecil dipilih sebagai *best node* dan dimasukkan ke $CLOSE$.

4. Fungsi Evaluasi (Melalui A dan F):

$$f(E)=9265$$

$$f(G)=9470$$

$$f(D)=11300$$

$$f(H)=9581$$

$$f(I)=9400$$

$$f(C)=g(F)+g(F \text{ ke } C)+h(C)=1600+3700+5300=10600$$

Titik D, E, G, dan I melalui *parent* A dan F

$$f(D)=g(F)+g(F \text{ ke } D)+h(D)=1600+1500+7550=9900$$

$$f(E)=g(F)+g(F \text{ ke } E)+h(E)=1600+14+7550=9164^*$$

$$f(G)=g(F)+g(F \text{ ke } G)+h(G)=1600+1200+7550=10770$$

$$f(I)=g(F)+g(F \text{ ke } I)+h(I)=1600+9800+0=11400$$

Successor F dibangkitkan, yaitu C, D, E, G, dan I. Titik D, E, G, dan I sudah pernah berada di $OPEN$, maka perlu dilakukan pengecekan *parent* dari masing-masing titik tersebut perlu diganti atau tidak. Berdasarkan perhitungan di atas *parent* dari titik G dan I tidak perlu diganti karena menghasilkan bobot yang lebih besar, sementara *parent* titik D dan E perlu diganti karena menghasilkan bobot yang lebih kecil. Langkah ini menghasilkan $OPEN=[E, G, D, H, I, C]$ dan $CLOSE=[A, B, C, F]$. Titik E dengan bobot terkecil dipilih sebagai *best node* dan dimasukkan ke $CLOSE$.

5. Fungsi Evaluasi (Melalui A, F, dan E):

$$f(G)=9470$$

$$f(D)=9900$$

$$f(H)=9581$$

$$f(I)=9400^*$$

$$f(C)=10600$$

Titik B, D, dan H melalui *parent* A, F, dan E

$$f(B)=g(E)+g(E \text{ ke } B)+h(B)=1614+2900+5210=9724$$

$$f(D)=g(E)+g(E \text{ ke } D)+h(D)=1614+1400+6800=9814$$

$$f(H)=g(E)+g(E \text{ ke } H)+h(H)=1614+3900+5000=10514$$

Successor E dibangkitkan, yaitu B, D, dan H. Titik B, D, dan H sudah pernah berada di $OPEN$, maka perlu dilakukan pengecekan *parent* dari masing-masing titik tersebut perlu diganti atau tidak. Berdasarkan perhitungan di atas *parent* dari titik B dan H tidak perlu diganti karena menghasilkan bobot yang lebih besar, sementara *parent* titik D perlu diganti karena menghasilkan bobot yang lebih sedikit. Langkah ini menghasilkan $OPEN=[G, D, H, I, C]$ dan $CLOSE=[A, B, C, F, E]$. Titik I dengan

bobot terkecil dipilih sebagai *best node* dan dimasukkan ke $CLOSE$.

Perhitungan dilanjutkan sampai langkah ke-21 dan kemudian didapatkan langkah selanjutnya sebagai berikut:

22. Fungsi Evaluasi (Melalui A, G, F, E, D, B, dan I):

$$f(D)=15995$$

$$f(F)=29181$$

$$f(H)=14795^*$$

Titik H melalui *parent* A, G, F, E, D, B, dan I

$$f(H)=g(I)+g(I \text{ ke } H)+h(H)=14614+9800+5000=29414$$

Successor I dibangkitkan, yaitu H. Titik H sudah pernah berada di $CLOSE$, akan tetapi melalui *parent* yang berbeda sehingga perlu dilakukan pengecekan ulang. Berdasarkan perhitungan di atas *parent* titik H tidak perlu diganti karena menghasilkan bobot yang lebih besar. Langkah ini menghasilkan $OPEN=[D'', F'', H'']$ dan $CLOSE=[A, B, C, F, E, I, G, H, D, C', B', F', E', B'', D', C'', H', I', B''', C''', I']$. Titik H'' dengan bobot terkecil dipilih sebagai *best node* dan dimasukkan ke $CLOSE$.

23. Fungsi Evaluasi (Melalui A, G, F, E, D, B, C, dan H):

$$f(D)=15995^*$$

$$f(F)=29181$$

$$f(I)=g(H)+g(H \text{ ke } I)+h(H)=9795+9800+0=19595$$

Successor H dibangkitkan, yaitu I. Titik I sudah pernah berada di $CLOSE$, akan tetapi melalui *parent* yang berbeda sehingga perlu dilakukan pengecekan ulang. Langkah ini menghasilkan $OPEN=[D'', F'', I'']$ dan $CLOSE=[A, B, C, F, E, I, G, H, D, C', B', F', E', B'', D', C'', H', I', B''', C''', I', H'']$. Titik D'' dengan bobot terkecil dipilih sebagai *best node* dan dimasukkan ke $CLOSE$.

24. Fungsi Evaluasi (Melalui A, G, F, E, B, C, dan D):

$$f(F)=29181$$

$$f(I)=19595^*$$

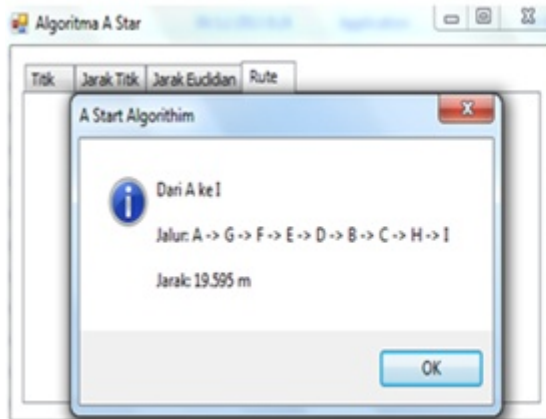
Titik I melalui *parent* A, G, F, E, B, C, dan D

$$f(I)=g(D)+g(D \text{ ke } I)+h(I)=8895+1300+0=21895$$

Titik I dengan bobot terkecil, yaitu 19595 terpilih sebagai *best node*. Karena *best node*-nya sama dengan *goal* dan telah memenuhi semua titik yang tersedia berarti solusi ditemukan. Proses ini menghasilkan rute terpendek yang dilalui adalah CV Mitra Adi

Busana - Hotel Gumaya - Hotel Merbabu - Hotel Novotel - Hotel Ibis - Hotel Grand Saraswati - Hotel Royal Phoenix - Hotel Neo Candi - Hotel Plaza dengan panjang rute terpendek 19595 meter. Hasil tersebut sama dengan perhitungan algoritma A* berbantuan *software* Visual Basic.Net.

Gambar 2 adalah tampilan hasil perhitungan dengan menggunakan *software* Visual Basic.Net:



Gambar 2 Tampilan Hasil Akhir *Software*

Simpulan

Berdasarkan hasil analisis yang dilakukan, diperoleh rute terpendek pendistribusian barang yaitu, CV Mitra Adi Busana - Hotel Gumaya - Hotel Merbabu - Hotel Novotel - Hotel Ibis - Hotel Grand Saraswati - Hotel Royal Phoenix - Hotel Neo Candi - Hotel Plaza dengan panjang rute terpendek 19595 meter. Hasil ini sesuai dengan perhitungan berbantuan *software* Visual Basic.Net. Berdasarkan penelitian di atas, dapat disimpulkan bahwa algoritma A* dapat digunakan untuk menyelesaikan masalah pencarian rute terpendek pendistribusian barang CV Mitra Adi Busana Semarang.

Daftar Pustaka

- Ananda, P.S. N., S. Wahjuni, & E.P. Giri. 2010. Penentuan Rute Terpendek Menggunakan Variasi Fungsi Heuristik Algoritme A* pada Mobile Devices. *Jurnal Ilmiah Ilmu Komputer*. 15(2). 17-24.
- Gusyanto. 2012. *Perancangan Aplikasi Penentuan Rute Angkutan Kota Berbasis Android Menggunakan Algoritma A* untuk Daerah Jakarta Barat*. Undergraduate Thesis. Jakarta: BINUS.

Hakim, L. 2012. *Pemrograman Visual Basic*. Jakarta: Universitas Bunda Mulia.

Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.

Setiyawan. 2012. *Optimasi Perjalanan Rute Ambulance Menggunakan Algoritma A Star*. Jurusan Teknik Industri: ITS. Tersedia di <http://digilib.its.ac.id>.

Supriyani. 2008. *Analisis Perancangan Sistem Informasi Geografi untuk Pencarian Rute Terpendek Kunjungan ke Sekolah-sekolah Menengah Umum dan Kejuruan di Jakarta Barat pada Universitas Bina Nusantara*. Undergraduate Thesis. Jakarta: BINUS.